Future Networking For Scalable I/O

H. Chen, J. Decker, and N. Bierbaum
Sandia National Laboratories
PO Box 969, Livermore, CA
USA
Hycsw,jcdecke,nrbierb@sandia.gov

ABSTRACT

Large clustered computers provide low-cost compute cycles, and therefore have promoted the development of sophisticated parallel-programming algorithms based on the Message Passing Interface. Storage platforms, however, fail to keep pace with similar advances. This paper compares standard 4X InfiniBand (IB) to 10-Gigabit Ethernet (GbE) for use as a common storage infrastructure in addition to message passing. Considering IB's native ability to accelerate protocol processing in hardware, the Ethernet hardware in this study provided similar acceleration using TCP Offload Engines. We evaluated their I/O performance using the IOZONE benchmark on the iSCSI-based TerraGRID parallel filesystem. Our evaluations show that 10GbE, with or without protocol-offload, offered better throughput and latency than IB to socket-based applications. Although protocol-offload in both 10GbE and IB demonstrated significant improvement in I/O performance, large amount of CPU are still being consumed to handle the associated data-copies and interrupts. The emerging RDMA technologies hold promises to remove the remaining CPU overhead. We plan to continue our study to research the applications of RDMA in parallel I/O.

KEY WORDS
InfiniBand, 10GbE, TOE, Parallel I/O, Cluster, HPC

1. Introduction

Commoditization of microprocessor and network technology has fostered an environment where large-clustered computers [1] can provide the same compute power as specialized Symmetric Multiprocessing Systems (SMP) [2], but at a tenth of the cost. This drop in cost-per-compute cycle has promoted the development of sophisticated parallel-programming algorithms based on the Message Passing Interface (MPI) [3]. Storage platforms, however, fail to keep pace with similar advances. Today's high-speed clusters can easily use the latest in interconnect technologies (e.g. InfiniBand [4]) for node-to-node communication, but the I/O is bottlenecked by Network File System (NFS) [5]. This paper compares standard 4X InfiniBand (IB) to 10-Gigabit Ethernet (GbE) [6], for use as a common infrastructure for storage in addition to message passing. Considering IB's native ability to accelerate protocol processing in hardware, the Ethernet hardware in the study provided similar acceleration using TCP Offload Engines (TOE) [7]. This study is unique because it concentrates on parallel I/O performance instead of message-passing.

In this study, the achievable aggregate bandwidth was measured using the TerraGRID parallel filesystem, developed by Terrascale Technologies [8]. Since TerraGRID uses iSCSI technology [9], it provided the necessary hooks for both interconnect to operate at full bandwidth. For IB, the SDP [10] interface was used to send SCSI control and data commands over its Reliable Connection Service. For 10-Gigabit Ethernet, the familiar socket interface was used to utilize TCP's reliable data transport. Organization of the paper is as follows: Background technologies used in the study are covered in Section 2; a description of testbed software and hardware components in Section 3; benchmark methodology in Section 4; results and analysis in Section 5; and, finally, conclusion and future plans in Section 6.

2. Background

Parallel applications combine the power of a large number of processors to solve a single problem. Applications that solve large science problems also move large amount of data, at fixed intervals, between memory and storage, requiring parallel I/O paths to satisfy High Performance Computing (HPC) demands. In addition, parallel applications that distribute their global data structure in distributed memory can greatly benefit from the ability to access separate portions of the same file at the same time. Parallel filesystem designed to allow concurrent accesses and provide parallel paths will greatly ease parallel code development, and significantly simplify the post-processing and analysis of large and complex scientific datasets.

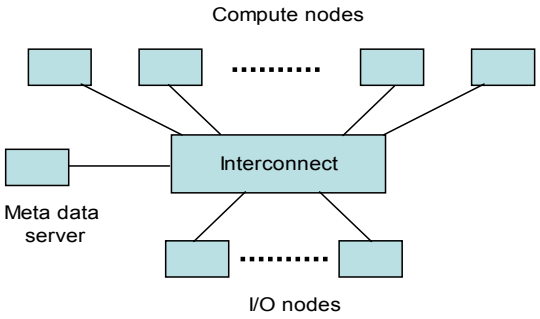


Figure 1, Parallel I/O in Clustering Computing

Figure 1 depicts a popular parallel I/O architecture [11] adopted in HPC environment. Depending on its performance requirement and the application profile, a FLOP-to-Byte/s ratio ranging from 50:1 to 500:1 are used as guidelines to calculate the ratio of Compute to I/O node in designing the cluster. This architecture offers *n* parallel I/O paths to access dedicated storage behind *n* I/O nodes. To allow concurrent accesses to different portions of the same file, the cluster's parallel filesystem presents a global view of the filesystem to all compute processors via a Meta-Data Service (MDS) for directory, filename, and data location lookup; by returning the resolved data location map to its compute clients, the cluster filesystem allows direct I/O operations between compute and I/O nodes in parallel.

Previous studies evaluated IB and 10 GbE as the message passing interconnect. This study is unique because it concentrates on parallel I/O performance. The technical background of the emerging technologies that are relevant is presented in the following paragraphs.

2.1 The TerraGRID Parallel Filesystem

TerraGRID is an iSCSI-based, block-level, scalable I/O platform, with its client software running on compute nodes and server code on I/O nodes. The iSCSI protocol is an IETF standard designed to encapsulate SCSI command and response in TCP/IP packets. It is created to reduce the total cost of ownership by leveraging the widely deployed IP infrastructure. Figure 2 illustrates the software components of TerraGRID. As shown, the TerraGRID platform fully harnesses Linux file systems and utilities: At start up, the TerraGRID iSCSI logic presents all of its server/targets to each client/initiator as SCSI devices; each client then uses the Linux MD driver to construct a software RAID over these target devices; finally, TerraGRID implements a Shared Access Scheduling Scheme (SASS) to enable generic Linux ext2 to act as a massively parallel filesystem. The TerraGRID filesystem maintains distributed meta-data on all targets. All I/O requests are parallelized by striping them across the RAID'ed devices, relying on the SASS algorithm to maintain data consistency between concurrent accesses from multiple processes.

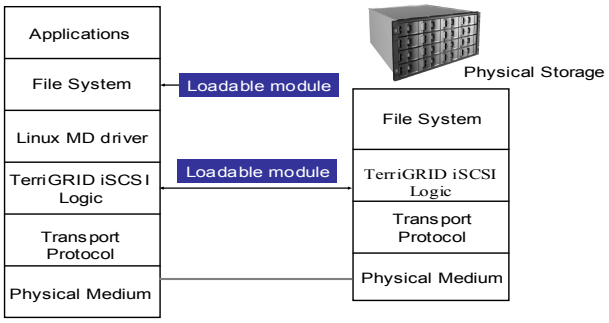


Figure 2, the TerraGrid Parallel I/O Platform

2.2 InfiniBand and the Mellanox Socket Direct Protocol

InfiniBand (IB) is an emerging high-speed, low latency interconnect technology by IBA [12]. IB is rapidly gaining popularity in the HPC communities because of its performance characteristic and the commodity pricing. This technology processes its protocol in hardware to minimize CPU overhead and achieve high throughput. In addition, IB supports Remote Direct Memory Access (RDMA) [13] that delivers data directly to remote application without interrupting the receiving processor. In fact, RDMA is the key to IB's impressive latency performance because it significantly reduces memory and CPU overhead needed otherwise to handle multiple data-copies and associated interrupts. This saving is critical to the application performance on processors with network speed of 10 Gigabit bits per second (Gbps) or greater, because advances in microprocessor and memory technologies have lagged behind those of networking in recent years.

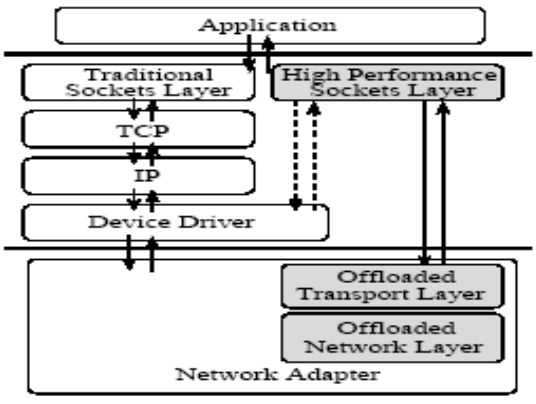


Figure 3, Host TCP/IB and Socket Direct Protocol over IB

IB uses two approaches to support existing TCP applications. The first adopts an overlay model that uses IB merely as the physical medium of an IP network

(IPoIB or TCP/IP-IB) [14]. The second implements a Direct Socket Protocol to provide TCP applications with the familiar Socket-like API, and uses the IB protocol to deliver reliable transport (SDP-IB). SDP is specifically designed to transparently support existing sockets-based applications and still sustain most of the performance benefit of IB. The IPoIB and SDP-IB protocol stacks are contrasted in Figure 3. As shown, SDP bypasses the host resident TCP/IP stack and relies on the hardware IB protocol for reliable data transport. The Mellanox implementation we used in our study defines SDP as a new AF_INET protocol family. Existing socket applications transparently connect to the SDP/IB protocol through an environmental variable.

2.3 10-Gigabit Ethernet (10 GbE) and the Chelsio TCP Offload Engine (TOE)

Due to ease of deployment and low cost, Ethernet (10, 100, and 1000 Mbps) remains the most prevalent networking technology in local area networks (LAN). We anticipate its ubiquity becoming even more prominent as long-haul network providers move away from the expensive SONET towards 10-Gigabit Ethernet. However, because of its performance drawback, Ethernet hasn't been widely adopted by the HPC communities as the cluster-interconnect. The emerging 10 GbE has the potential to bridge this performance gap, but is not competitively priced yet today. The market trend in Figure 4 demonstrates, however, that the manufacturing volume of 10 GbE has already reached the level to drive its costs down exponentially, promising similar commodity cost advantage offered by InfiniBand in the near future.

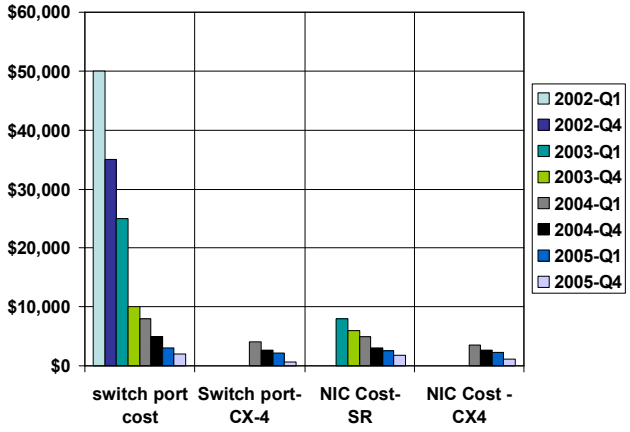


Figure 4, 10-Gigabit Ethernet Trend

The TCP-offload-Engine (TOE) is another attempt to reduce the CPU's burden in processing TCP/IP to deliver throughput at 10 Gbps. Chelsio is one of the early adopters that implement the TOE technology over 10 GbE (TOE-10GbE). Unlike the SDP-IB, the software design of Chelsio's TOE retains Linux kernel's existing sockets layer. Based on a pre-configured policy base, the sockets layer pushes the processing of TCP either to the Chelsio offload engine or the host-stack (see Figure 5). In the latter case, the TOE device is simply used as a regular network interface card. The Chelsio software architecture [15] consists of two major components: A TCP offload module (TOM) and an offload driver. TOM is the upper layer of the software TOE stack; it implements a subset of its own transport-layer API in order to support portions of TCP that cannot be processed on the TOE hardware. In addition, TOM is responsible for the maintenance of the state of all offloaded connections. The offload driver is the lower layer of the software TOE stack. It is responsible for direct manipulation of TOE and its associated resources.

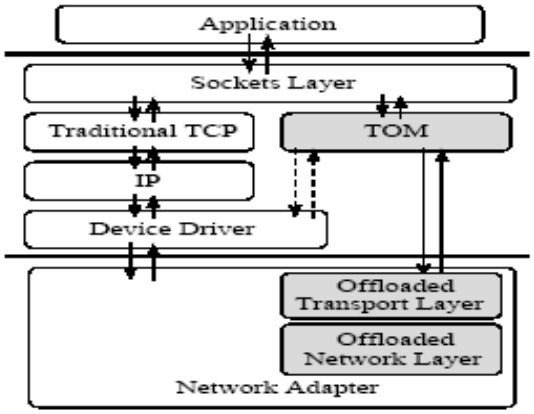


Figure 5, Host TCP/IP and TOE over 10 Gigabit Ethernet

3. Testbed Configuration

For this study twelve dual Opteron systems were used, four as the TerraGRID servers (or targets) and the remaining as the TerraGRID clients (or initiators). Table 1 lists the node configuration and Figure 6 depicts the testbed topology.

Table 1, Dual Opteron Node Configuration

Operating System	Gentoo 2005.0 (kernel v. 2.4.25)
Parallel Filesystem	TerraGRID v. 1.0, Terrascale Technologies
Motherborad	Tyan S2895A
Processor	Opteron (SKT940 2.2 GHz)
Memory	2GB on client and 8 GB on server (ATP 1GB PC3200)
10-Gigabit Ethernet TCP-Offload-Engine (TOE)	Chelsio T210 10BaseX (rev 3)
InfiniBand Host Bus Adapter (HBA)	Mellanox Technologies MT23108

Please note that we had to use the 100 Mhz PCI-X slots for our InfiniBand HBA and 10-Gigabit Ethernet TOE in order to avoid bug 56 in the AMD 8131 133 MHz PCI-X Hyper Transport bridge.

Because our goal is to evaluate I/O network technologies, we configured RAM disks using the Linux TMPFS on the targets in order to eliminate disk I/O considerations from our performance evaluation. We ran oneSIS [16] on the headnode to boot the rest of our cluster nodes; "oneSIS" is a Sandia developed open source cluster management package. Other key software components that we used include the Mellanox IB SDP stack and the Chelsio TOE kernel module and device driver.

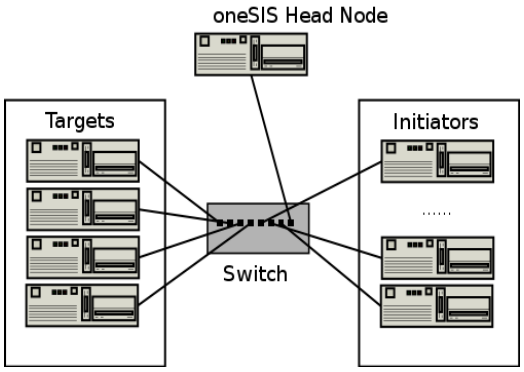


Figure 6, the Testbed Topology

4. Benchmark Methodology

4.1 The Custom Benchmark Framework

A custom benchmark framework was developed to integrate the definition, execution, and to organize the results and related information. This framework uses XML definition files to define the test environment, the test program parameters, and the scheduling of simultaneous runs across multiple hosts. Results of each run are reported in a series of XML, HTML and serialized compressed data files to allow easy reviewing and consistent, unambiguous searching and processing. Currently, the framework consisted of Iozone [17], Netperf [18], and a custom tool on file system operations. IOzone is a filesystem benchmark tool that generates and measures a variety of file operations, and Netperf a popular tool used to measure the throughput and latency of different types of networking technologies.

In addition to recording benchmark results, this framework also launches concurrent remote control processes to record the system resource usage during each test run. We have also developed post processing tools that convert data specific to a test type into a spreadsheet for further plotting and analysis.

4.2 Description of Test Suites

Three test scenarios were designed and performed separately for each of the 4 fabric technologies. Using Netperf, the first test was designed to baseline the performance characteristics of the individual technologies: "*TCP/IP-IB*", "*SDP-IB*", "*TCP/IP-10GbE*", and "*TOE-10GbE*". Our second test, also using Netperf, was designed to profile the fabric's scalability characteristics. Finally, we evaluated their parallel I/O performance using the Iozone benchmark over the TerraGRID parallel filesystem. For the second test, we configured the Netperf socket connections based on the parallel I/O profile used in TerraGRID (see Figure 7).

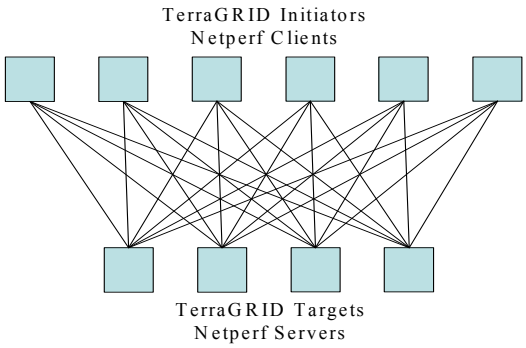


Figure 7, TerraGRID Parallel I/O Socket Connection Profile

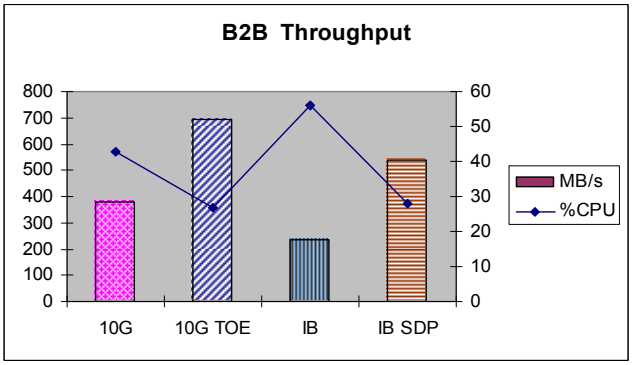
5. Result and Analysis

Before we launched the aforementioned test scenarios, we benchmarked TMPFS, the Linux RAM disk, using Iozone in order to set the upper limit of I/O throughput for our study. The IOzone I/O results are listed in Table 2.

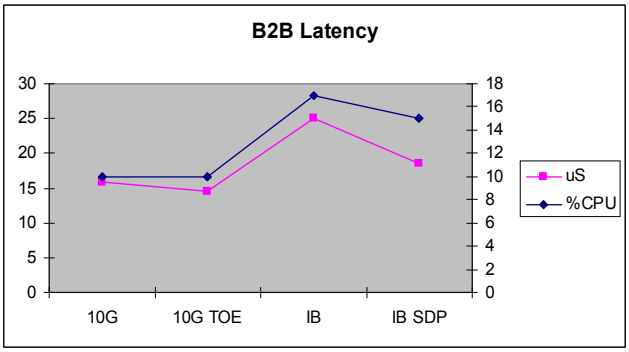
Table 2, IOZONE Results for TMPFS

File Size (GB)	Record Len (MB)	Write (KB/s)	Rewrite (KB/s)	Read (KB/s)	Reread (KB/s)
2	16	737,168	767,921	779,946	802,403

5.1 Technology Baseline



(a)



(b)

Figure 8, Back-to-back Netperf: (a) throughput and (b) latency

The Netperf results of our back-to-back tests are plotted in Figure 8. Figure 8a demonstrates that protocol offload (POE) in both IB and 10GbE performed better than their host stack counterpart with respect to throughput as well as CPU overhead. Similar performance advantages are also observed in the latency study (Figure 8b); POE in both IB and 10GbE again delivered better latency than their corresponding counterpart. In addition, the results show that the two 10GbE-based fabrics achieved better throughput and latency than their IB equivalent; for example, *TOE with 10GbE* outperformed SDP with IB, and *host TCP/IP with 10GbE* outperformed *host TCP/IP with IB*.

5.2 Fabric Scalability Baseline

We baseline the scalability of all four fabric configurations using concurrent Netperf sessions that follow TerraGRID's parallel I/O socket profile (see Figure 8), with each active client launching 4 concurrent Netperf sessions, 1 to each of the 4 servers. Figure 9 plots the aggregate throughput as a function of concurrent Netperf sessions launched from 1 to 6 clients. Again we show that *10 GbE with TOE* performed best and achieved linear scale up. But its aggregate throughput leveled at 4 concurrent hosts because we are bandwidth limited by the 4 I/O servers. *IB with SDP* achieved slightly better throughput than *10 GbE with host TCP/IP* initially, but scaled worse when additional load were introduced. *IB with host TCP/IP* delivered the worst aggregate throughput, although was able to achieve linear scale up, demonstrating that, at this low level of throughput, fabric bandwidth is not the performance bottleneck.

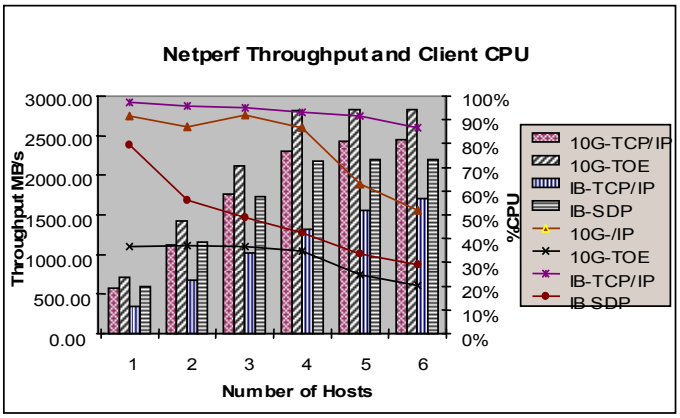


Figure 9, Fabric throughput baseline using Netperf

Figure 10 plots the end-to-end latency on each client reported by the Netperf test suites designed to baseline fabric scalability of the 4 technologies. Again, *10 GbE with TOE* showed better latency results as well as good scalability; there are only slight increases in latency as the number of concurrent hosts increased. *10 GbE with host TCP/IP* also performed surprisingly well. The anomaly shown on its single host data-point is a result, we believe, of the interrupt coalescing mechanism implemented on the Chelsio network card; with low traffic volume, Netperf's 64-byte messages used to measure latency were holdup by the interrupt coalescing scheme on the receiving card causing delayed delivery and consequently bigger latency values. We observed, quit unexpectedly, that both *IB with SDP* and *IB with host TCP/IP* displayed worse latency values than their *10 GbE* equivalent. In the case of *IB with SDP*, we also observe a steeper increase in latency after 4 concurrent hosts reflecting scalability problems, a phenomenon also observed in a separate study by Feng, et al [19]. We suspect this is an implementation issue; we were constrained to run an earlier SDP implementation because of TerraGRID's Linux 2.4.25 kernel dependency. We plan to repeat this test with an SDP developed by the Open IB Consortium [20] running on the Linux 2.6.12 kernel when the software becomes available.

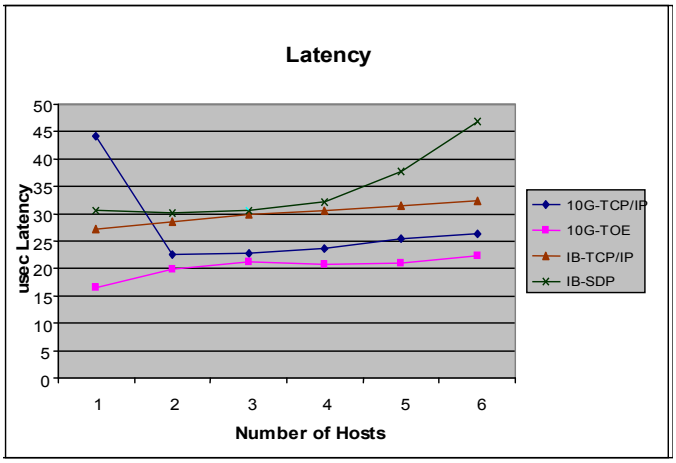


Figure 10, Fabric latency baseline using Netperf

5.3 TerraGRID IOZONE Benchmark

Figure 11 presents 4 plots, each reporting the results of a suite of Iozone aggregate-throughput tests of a separate technology. We arranged these plots for ease of comparison: the 2 *10-GbE* graphs on the left and the 2 *IB* on the right; the 2 *host TCP/IP* graphs on the top and the *TOE and SDP* at the bottom. From top to bottom, we observe that protocol-offload on both *10 GbE* and *IB* improved the filesystem performance by roughly 25%. From left to right, we found that the *10 GbE*-based technologies performed about 20% better than their *IB*-based counterpart. For all technologies, reads achieved better performance than writes; presumably because write operations incur more overhead due to their continuous need to extend storage allocation as the I/O proceeds.

Figures 12 shows the corresponding CPU overhead on individual client and server running the same suites of Iozone tests mentioned above. Figure 12a reveals that the CPU overhead on each client had actually declined with increasing number of concurrent sessions, corresponded

to their slight decline in throughput (Figure 11) most likely due to increased workload on servers. The server CPU load graph depicted in Figure 12b confirmed our deduction; the CPU overhead on servers had indeed increased proportionally against increasing concurrent sessions. As shown, differences in CPU overhead between the 4 configurations are within 5 to 15%, with IB-SDP being the most efficient followed by 10 GbE-TOE, IPoIB, and 10GbE. We also noticed that protocol offload for both IB and 10GbE represent only 5-10% of savings on CPU, suggesting that significant resources were still consumed to handle data-copies between kernel and user buffers and their associated interrupts. We plan to repeat our test when parallel filesystem implements RDMA. TCP RDMA, the iWARP protocol stack, is currently being drafted by the IETF Transport Area Workgroup. We believe the I/O performance will be further improved, and most importantly, RDMA would eliminate significant CPU overhead, thereby accelerating the execution of parallel applications that clusters are designed for.

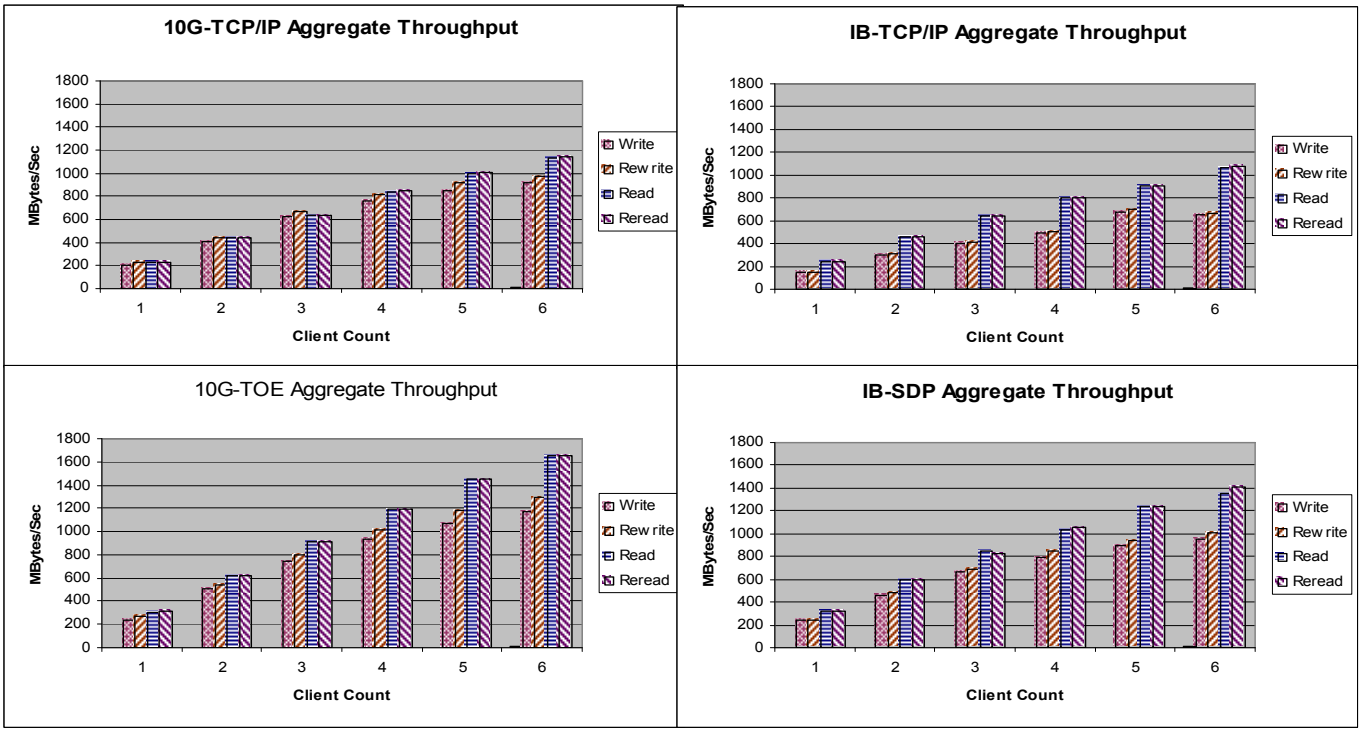
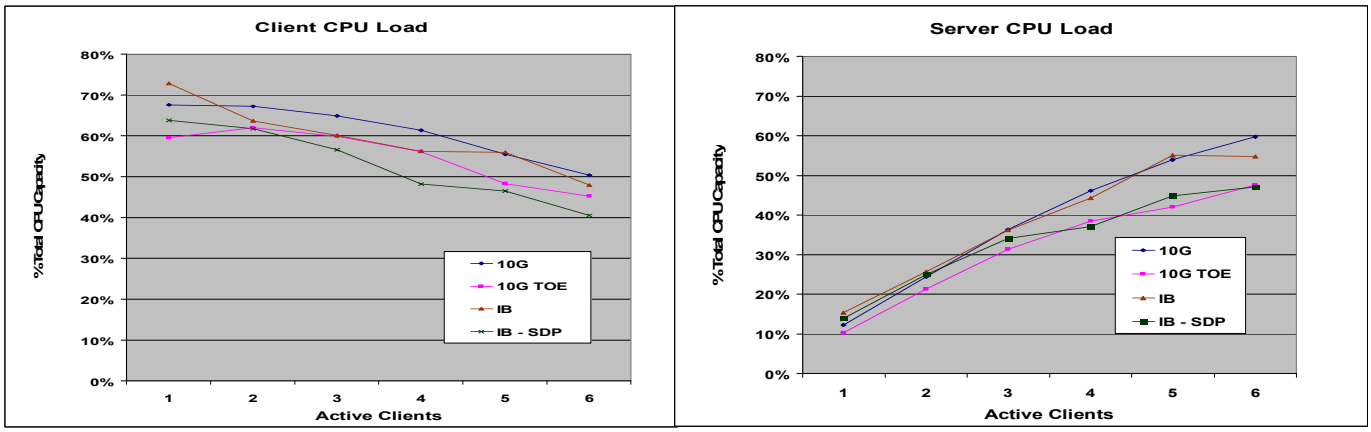


Figure 11, TerraGRID IOZONE Throughput for: (a) 10GbE-TCP/IP (b), IB-TCP/IP (c), 10GbE-TOE (d), and IB-SDP



(a) (b)

Figure 12, TerraGRID IOZONE CPU overhead on: (a) Client and (b) Server

6. Conclusion

Commoditization of microprocessor and network technology has fostered an environment where large-clustered computers can provide the same compute power as specialized Symmetric Multiprocessing Systems, but at a much lower cost. Although low cost in compute cycles has promoted the development of sophisticated parallel algorithms, storage platforms fail to keep pace with similar advances.

Gigabit Ethernet-based network has significant drawbacks in performance compared to special purpose networks such as InfiniBand. As such, Gigabit Ethernet has not been widely adopted in HPC as the cluster-interconnect, in spite of its e of deployment and compatibility with the ubiquitous Ethernet infrastructure. But the advent of 10 Gigabit Ethernet and TOE promises to bridge the performance gap. This study compared standard 4X InfiniBand (IB) to 10-Gigabit Ethernet, for use as a common infrastructure for storage and message passing. Considering IB's native ability to accelerate protocol processing in hardware; the Ethernet hardware in this study provided similar acceleration using TCP Offload Engines (TOE).

The evaluations show that in all four experimental scenarios, with and without protocol offload, 10GbE provides better performance than IB, demonstrating, perhaps, that because 10GbE and TOE are native to sockets based applications, they can offer better performance. This statement is not conclusive because it was necessary to use an earlier implementation of IB software due to TerraGRID's Linux kernel restriction. Nevertheless, this observation is significant to 10GbE and TOE manufacturers, because sockets interface is the most widely used interface for grids, file systems, storage, and other commercial applications. We believe, by leveraging on the mature IP technologies, 10GbE with TOE is a good candidate to implementing a scalable, sharable storage subsystem to meet the I/O demands of large parallel platforms. In addition, the market trend shown in Figure 4 indicates that the manufacturing volume of 10GbE has already reached the level to drive its costs down exponentially, promising similar commodity cost advantage offered by InfiniBand in the near future.

Although protocol-offload in both 10GbE and IB demonstrated significant improvement in I/O performance, we observe that large amount of CPU resources are still being consumed by I/O operations. The emerging RDMA technologies hold promises to remove the remaining CPU overhead from servicing data copies and associated interrupts. We plan to continue our study to research the applications of RDMA in parallel I/O when RDMA-based parallel filesystems become available.

References

[1] René J. Chevance, *Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions*, ELSEVER Digital Press, 2005

[2] D. E. Culler, J. P. Singh, *Parallel Computer Architecture, a Hardware/Software Approach*, Morgan Kaufmann Publishers, Inc., 1999, pp 269-271

[3] W. Gropp, E. Lusk, A. Skjellum, Using *MPI: Portable Parallel Programming with the Messaging-Passing Interface*, 2nd. Edition, the MIT Press, 1999

[4] W. T. Futral, *InfiniBand Architecture: Development and Deployment, a Strategic Guide to Server I/O Solutions*, Intel Press

[5] W. R. Stevens, *TCP/IP Illustrated Volume 1: The Protocols*, *Addison-Wesley Professional Computing Series,* 1994, pp 461-480, 542

[6] *10 Gigabit Ethernet Technology Overview*, IntelPRO Network Connections, http://www.intel.com/network/connectivity/resources/doc_library/white_papers/pro10gbe_lr_sa_wp.pdf,

[7] *From Ethernet Ubiquity to Ethernet Convergence: The Emergence of the Converged Network Interface Controller,* March, 2005, http://download.microsoft.com/download/3/5/0/3508460c-7bc2-4f7f-b5b9-4aefb981689b/C-NIC-WP102-R.pdf

[8] The TerraGRID Overview, http://www.terrascale.com/prod_over_e.html

[9] J. L. Hufferd, *iSCSI: The Universal Storage Connection, Addison-Wesley*, 2003

[10] P. Balaji, S. Narravula, K. Vaidyanathan, S. Krishnamoorthy, J. Wu, and D. K. Panda, *Sockets Direct Protocol over InfiniBand in Clusters: Is it Beneficial?* In ISPASS '04

[11] J. M. May, *Parallel I/O for High Performance Computing*, Morgan Kaufmann Publishers, 2001

[12] http://www.infinibandta.org/home

[13] R. Recio, et al, *An RDMA Protocol Specification (Version 1.0)*, Octobrt, 2002. http://www.rdmaconsortium.org/home/draft-recio-iwarp-rdmap-v1.0.pdf

[14] Vivek Kashyap, IP *over InfiniBand(IPoIB) Architecture*, April 2004, http://www.ietf.org/html.charters/ipoib-charter.html

[15] http://www.gridtoday.com/04/1206/104373.html

[16]Josh England, http://www.oneSIS.org

[17] Iozone Filesystem Benchmark, http://www.iozone.org/docs/IOzone_msword_98.pdf

[18] Netperf Network Benchmark Tool, http://www.netperf.org/netperf/training/Netperf.html

[19] W. Feng, et al, *Head-to-TOE Evaluation of High-Performance Sockets over Protocol Offload Engine,* Technical Report Los Alamos National Laboratory (LA-UR-05-4148), Ohio State University (OSU-CISRC-5/05-TR35)

[20]The Open IB Consortium, http://openib.org/

Acknowledgements

The following people deserve credit for helping with the success of this paper: Eric Van De Vreude for the compilation of data; Mitch Sukalski for the review and technical input; Felix Marti and John Thuotte for the tuning of Chelsio TOE; Tim Wilcox and Steeve McCauley for the support of TerraGRID.